(12) UK Patent Application (19) GB (11) 2 359 439 (13) A

(43) Date of A Publication 22.08.2001

(21) Application No 0030585.4

(22) Date of Filing 15.12.2000

(30) Priority Data
(31) 09477911 (32) 05.01.2000 (33) US

(71) Applicant(s)
International Business Machines Corporation
(Incorporated in USA - New York)
Armonk, New York 10504, United States of America

(72) Inventor(s)
Norman Hass

(74) Agent and/or Address for Service
IBM United Kingdom Limited
Intellectual Property Department, Hursley Park,
WINCHESTER, Hampshire, SO21 2JN,
United Kingdom

(51) INT CL^7
H04N 7/00

(52) UK CL (Edition S)
H4F FGXX

(56) Documents Cited
US 5847703 A US 5796439 A

(58) Field of Search
UK CL (Edition S) H4F FEP FEX FGXX FRS
INT CL^7 H04N 5/44 5/783 5/915 7/01
Online: WPI, EPODOC, JAPIO, TDB

(54) Abstract Title

Extracting frames representative of a shot from a sequence of frames

(57) A method for obtaining a representative sequence of frames 340E from a sequence of shot frames 200 comprising the steps of extracting frames from the sequence at a selection interval and storing the selected frames in a buffer 340A and then, when the buffer is full, culling frames from the buffer at a buffer interval and increasing the selection interval by multiplying the original selection interval by the buffer interval. Preferably after culling frames for the buffer the remaining frames are compacted 340C to retain the order of the selected frames. These steps are repeated so that a representative sequence 340E of frames is obtained, with the selected frames being spaced at regular intervals in the original shot sequence, by processing the sequence only once and without knowing the length of the sequence in advance.

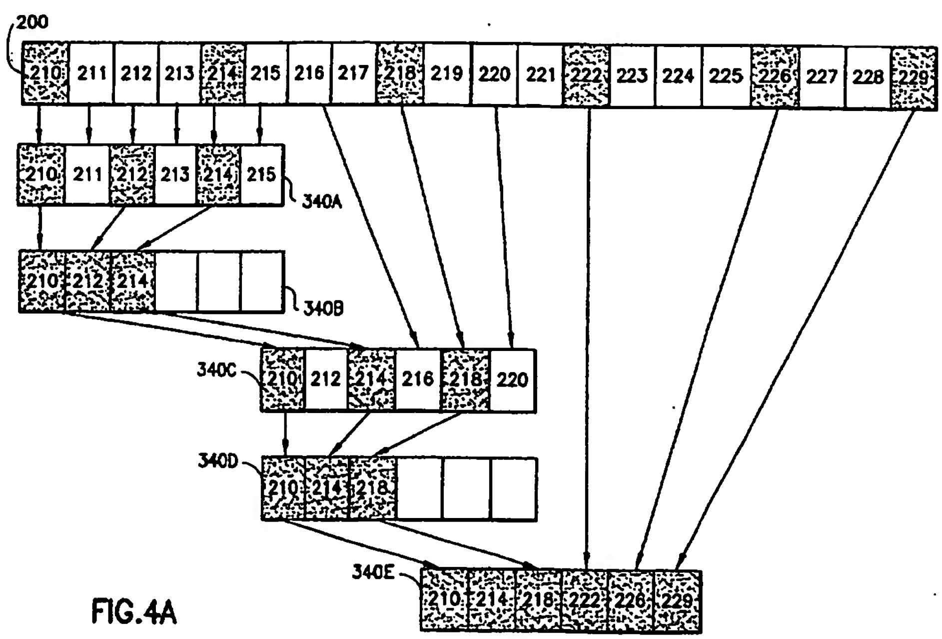


FIG.4A

GB 2 359 439 A

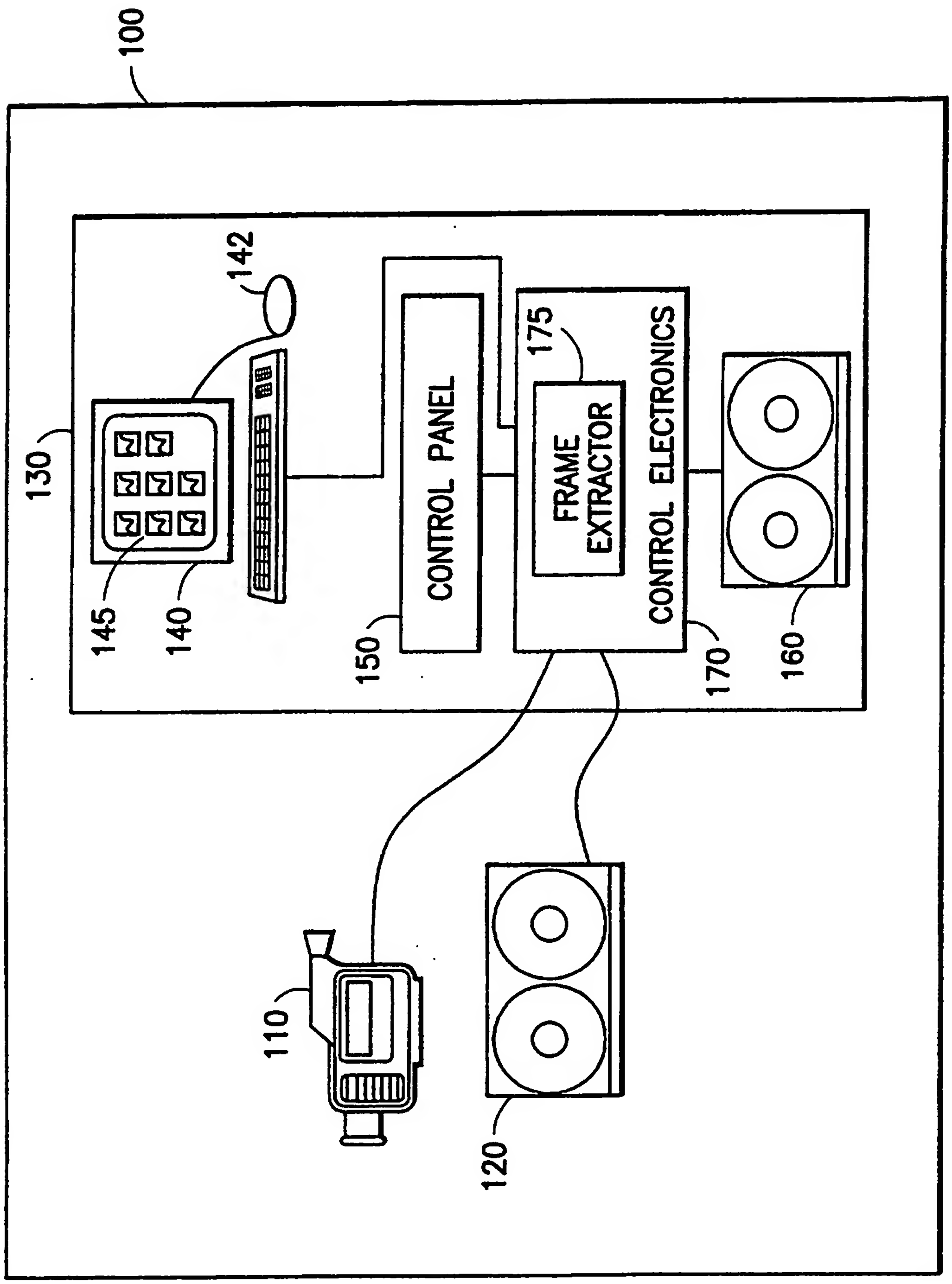

100
130
142
145
140
150
CONTROL PANEL
175
FRAME EXTRACTOR
CONTROL ELECTRONICS
170
160
110
120

FIG.1

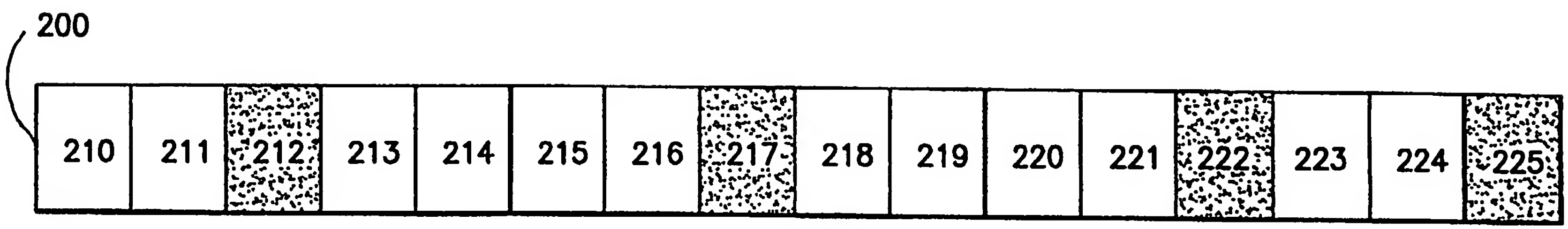

200
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225

FIG.2A

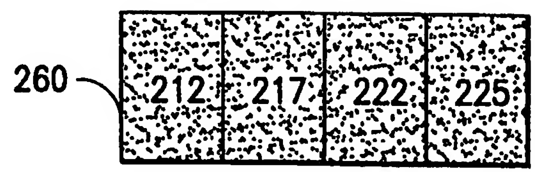
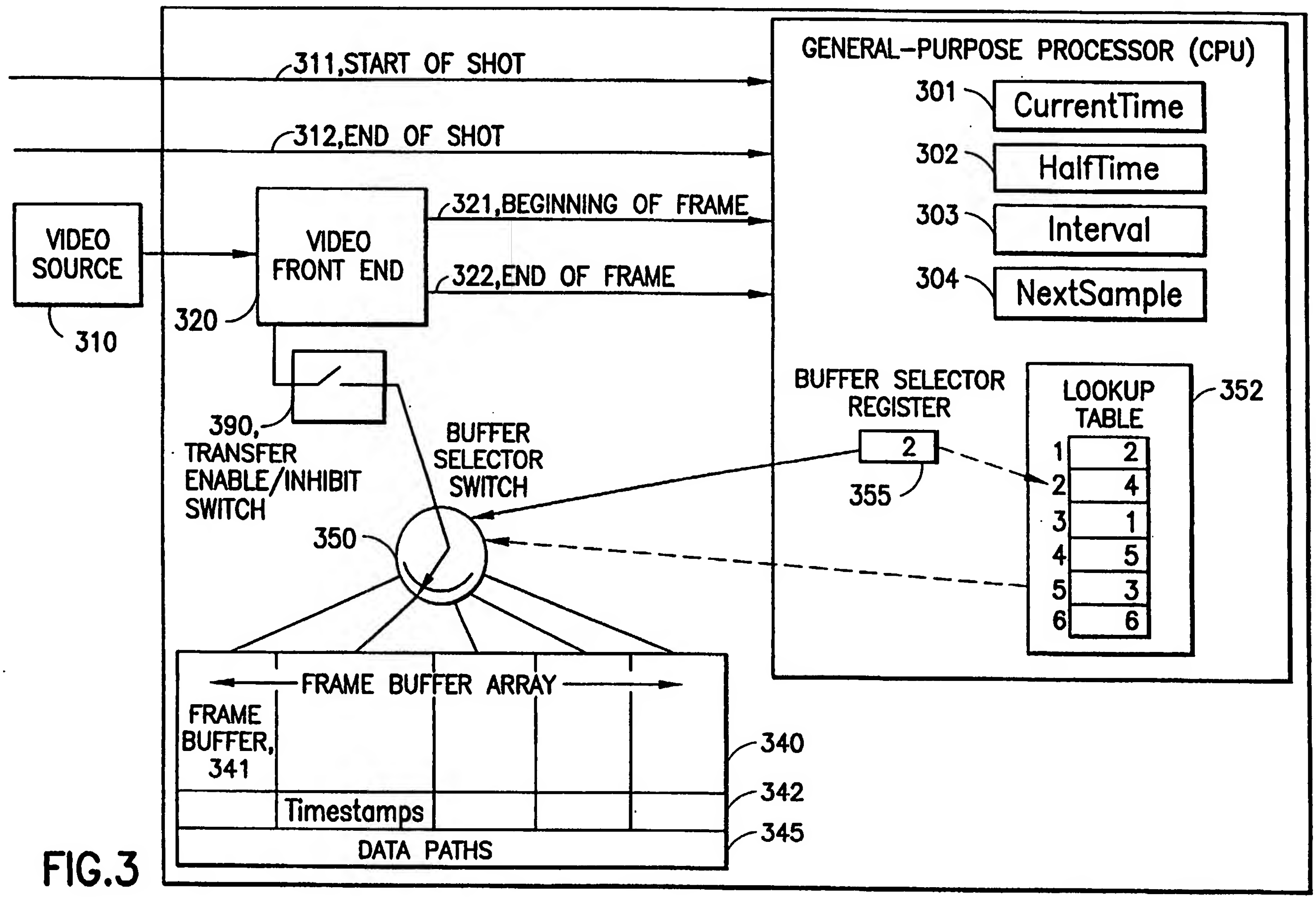

260
212
217
222
225

FIG.2B



GENERAL-PURPOSE PROCESSOR (CPU)
301
CurrentTime
302
HalfTime
303
Interval
304
NextSample
311,START OF SHOT
312,END OF SHOT
VIDEO SOURCE
310
VIDEO FRONT END
320
321,BEGINNING OF FRAME
322,END OF FRAME
BUFFER SELECTOR REGISTER
2
355
LOOKUP TABLE
352
1 2
2 4
3 1
4 5
5 3
6 6
390, TRANSFER ENABLE/INHIBIT SWITCH
BUFFER SELECTOR SWITCH
350
FRAME BUFFER ARRAY
FRAME BUFFER, 341
Timestamps
DATA PATHS
340
342
345

FIG.3

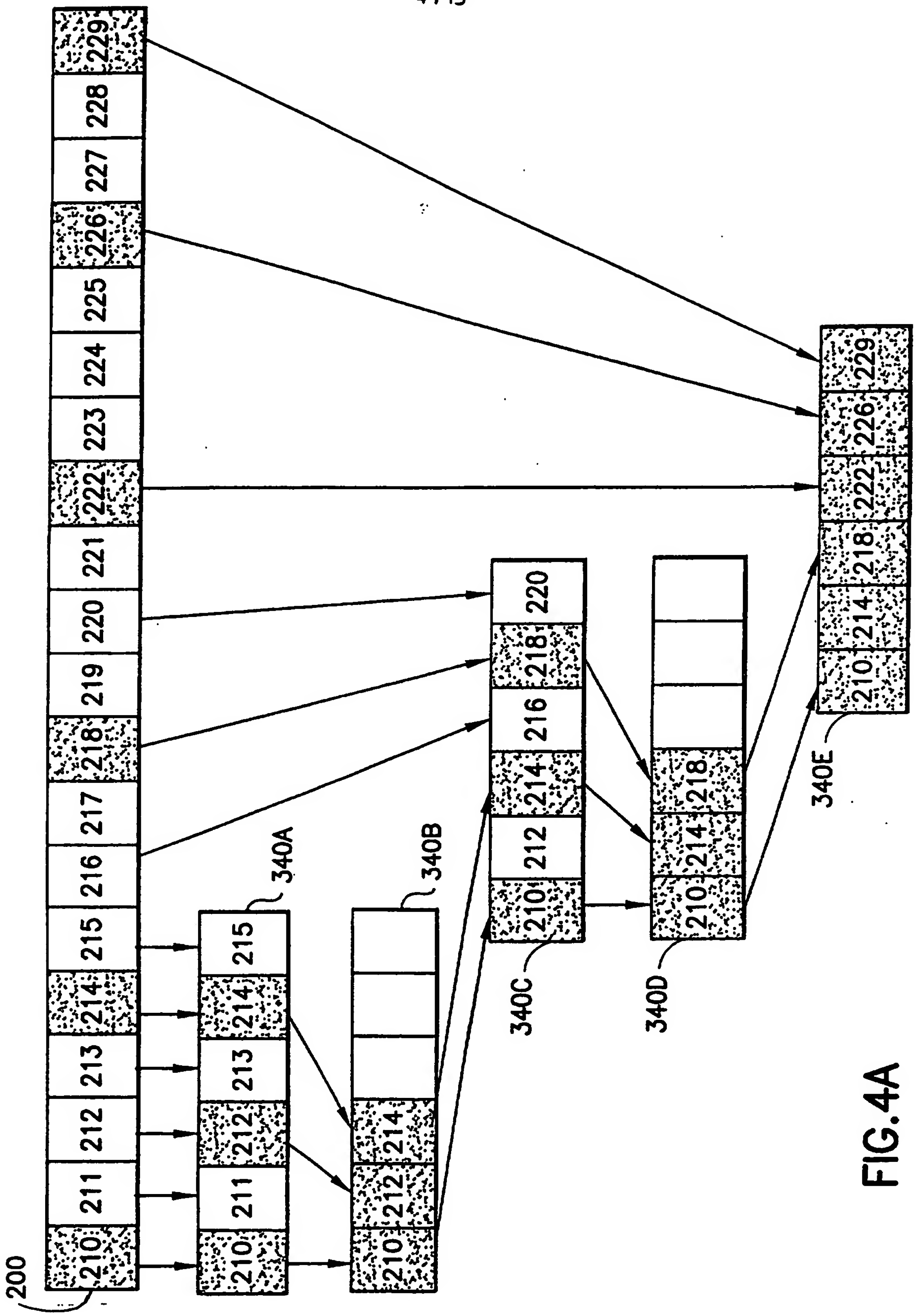
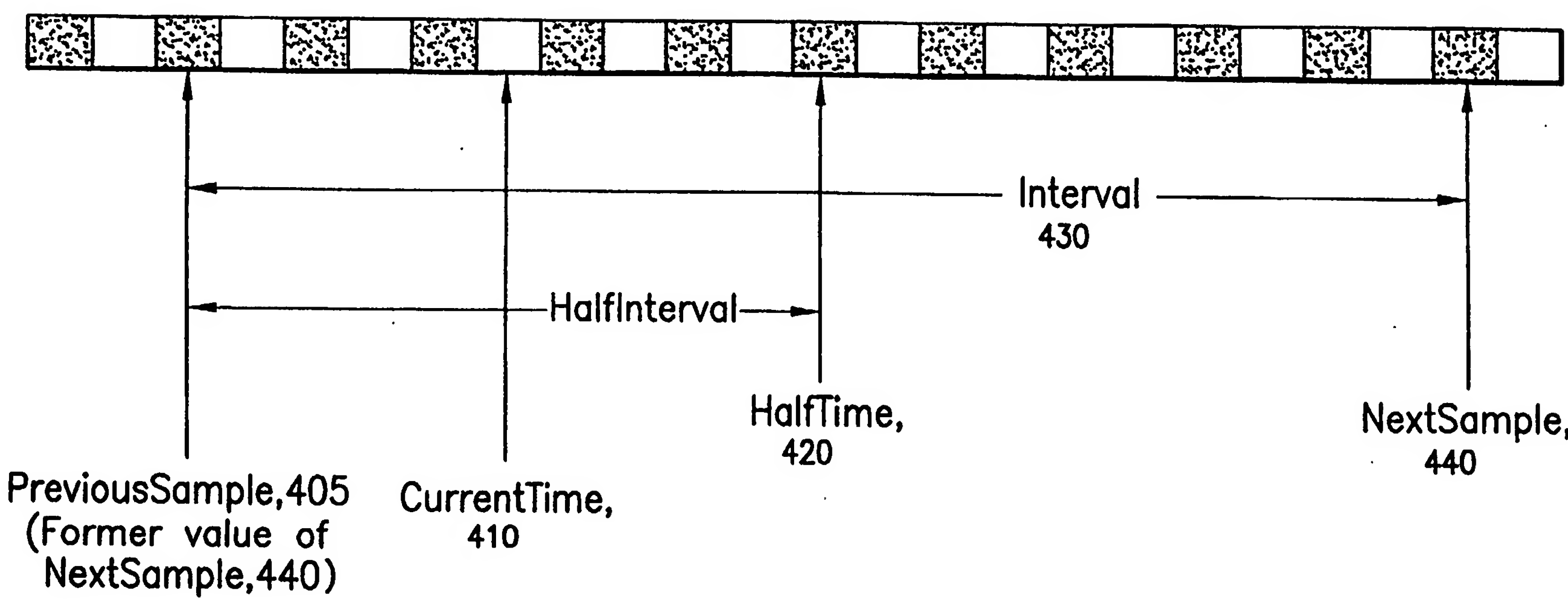

200
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
340A
340B
340C
340D
340E

FIG.4A



Interval
430
HalfInterval
PreviousSample,405
(Former value of
NextSample,440)
CurrentTime,
410
HalfTime,
420
NextSample,
440

FIG.4B

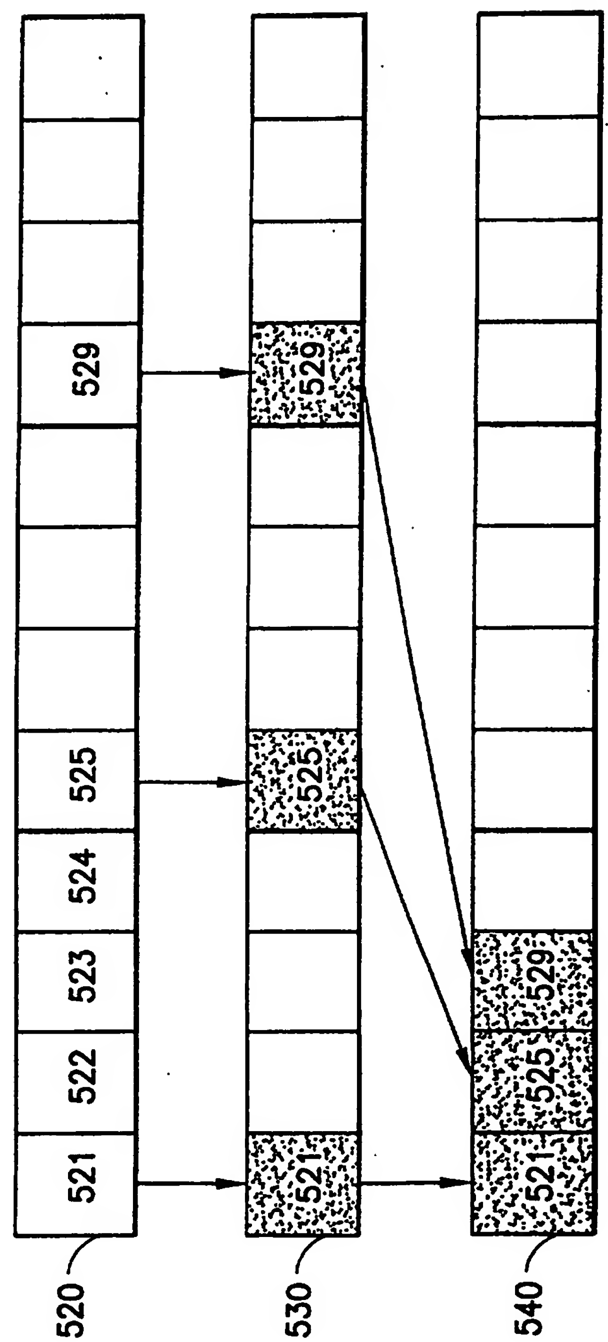
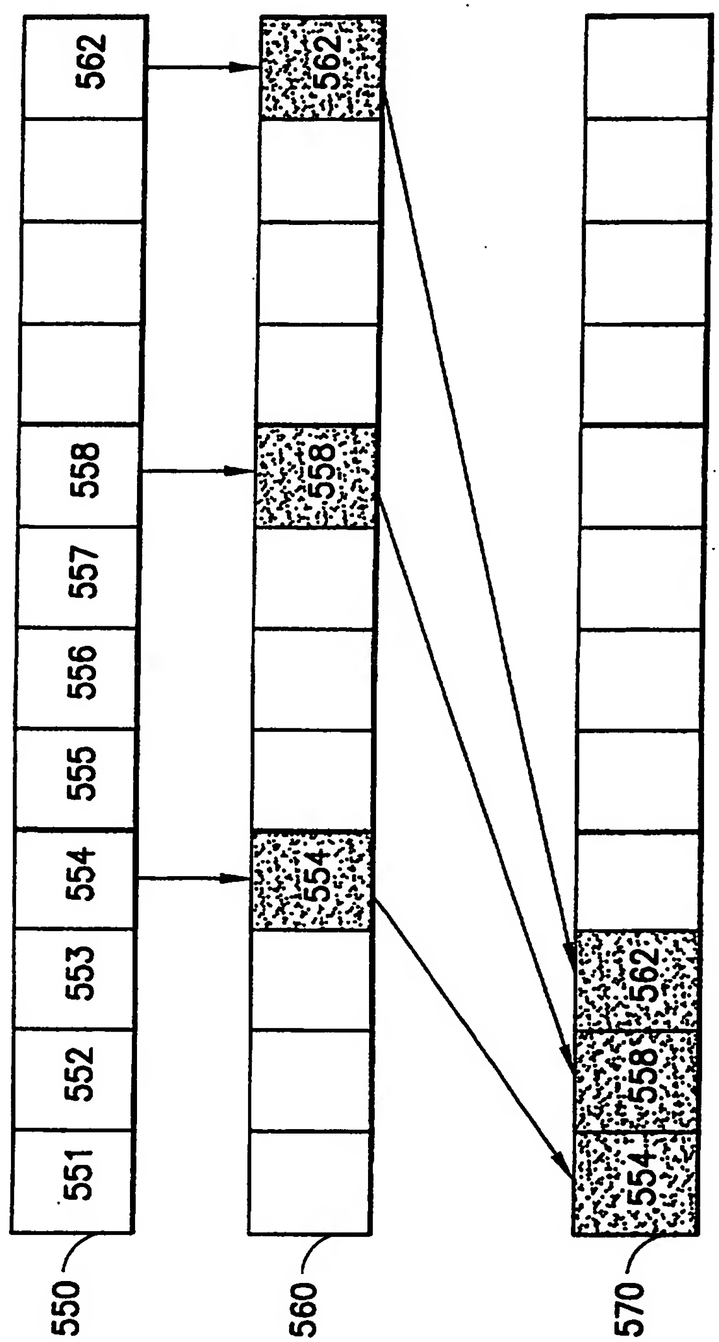


FIG.5A



562
558
557
556
555
554
553
552
551
550
562
558
554
560
562
558
554
570

FIG.5B

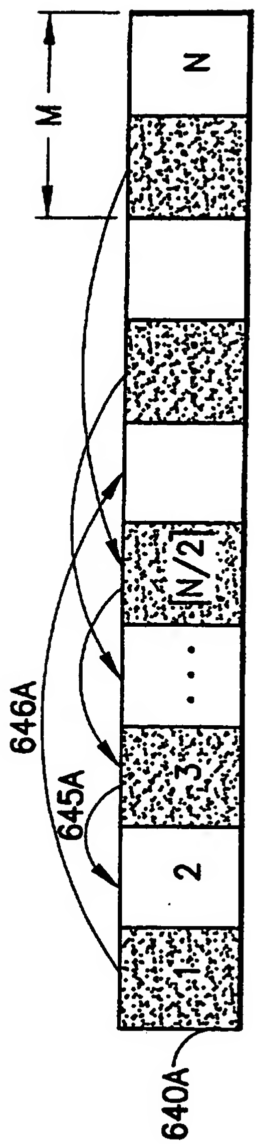


FIG.6A

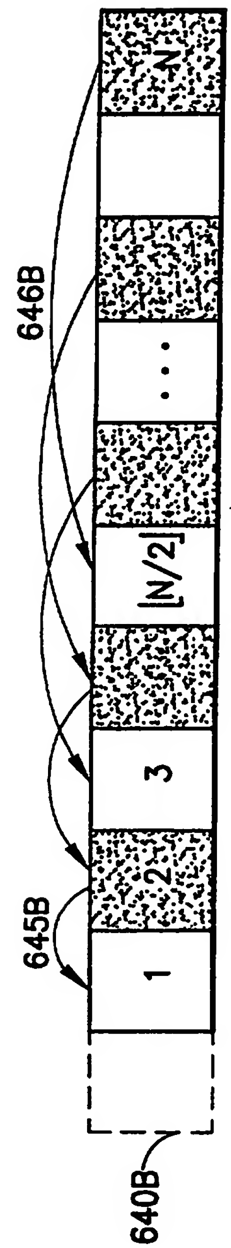


FIG.6B

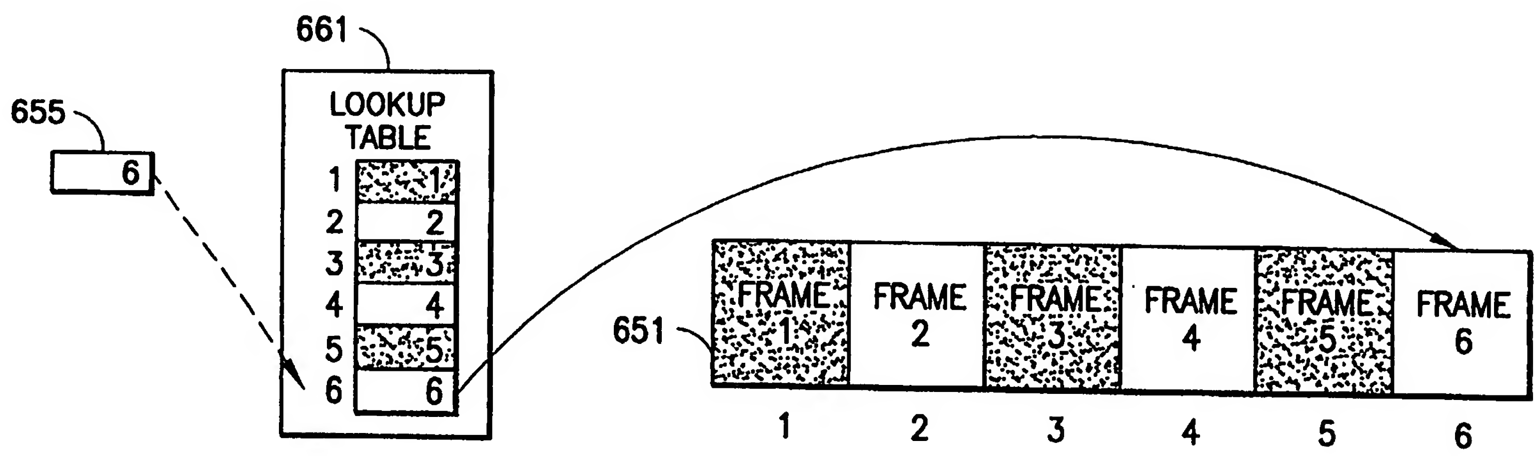

661
655
6
LOOKUP TABLE
1 1
2 2
3 3
4 4
5 5
6 6
651
FRAME 1
FRAME 2
FRAME 3
FRAME 4
FRAME 5
FRAME 6
1 2 3 4 5 6

FIG.6C

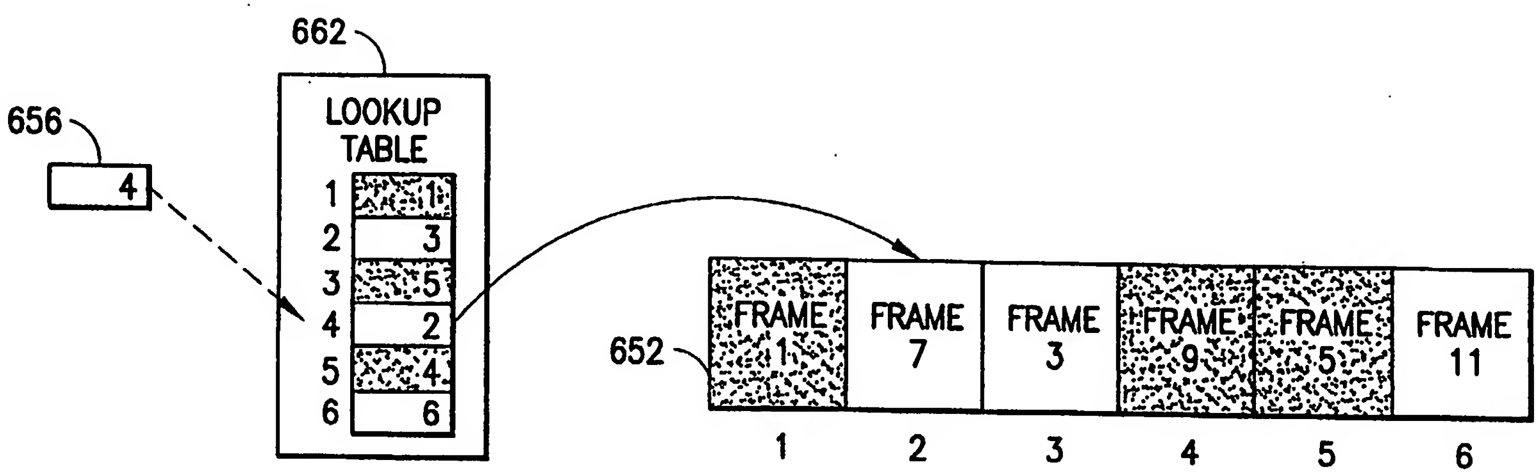

662
656
4
LOOKUP TABLE
1 1
2 3
3 5
4 2
5 4
6 6
652
FRAME 1
FRAME 7
FRAME 3
FRAME 9
FRAME 5
FRAME 11
1 2 3 4 5 6

FIG.6D

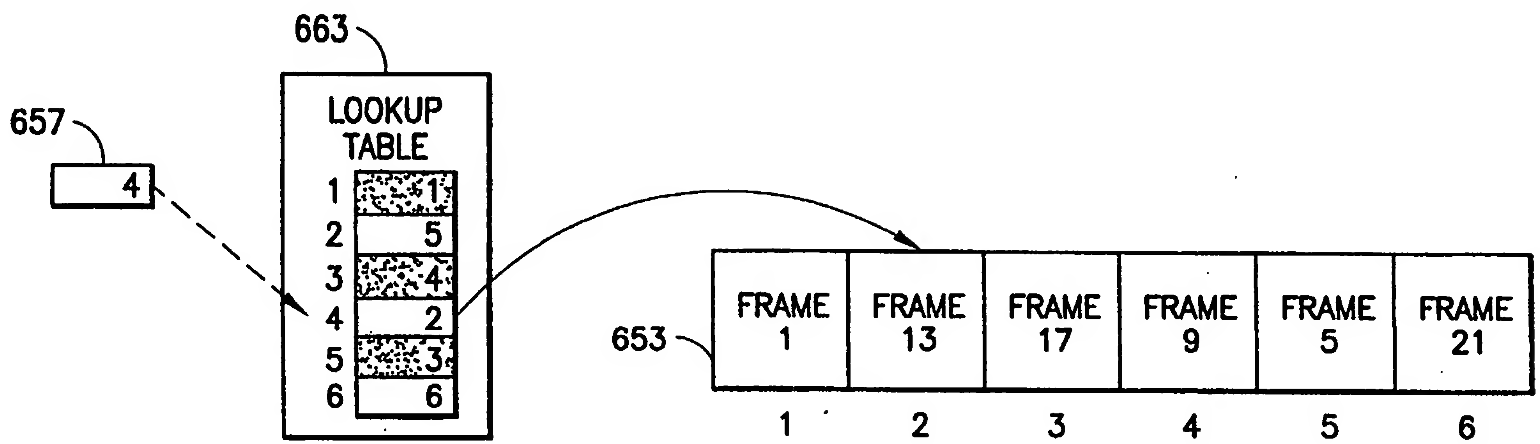

657
4
663
LOOKUP TABLE
1
2
3
4
5
6
1
5
4
2
3
6
653
FRAME 1
FRAME 13
FRAME 17
FRAME 9
FRAME 5
FRAME 21
1
2
3
4
5
6

FIG.6E

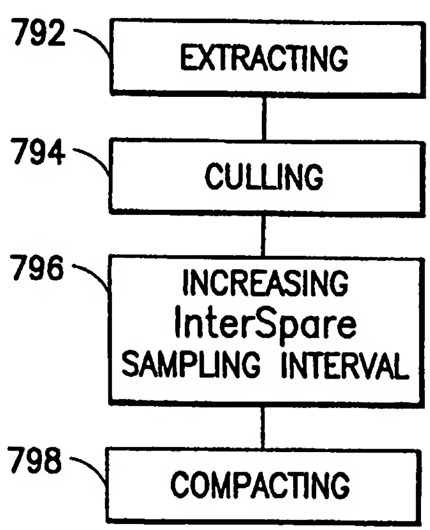

792
EXTRACTING
794
CULLING
796
INCREASING
InterSpare
SAMPLING INTERVAL
798
COMPACTING

FIG.7A

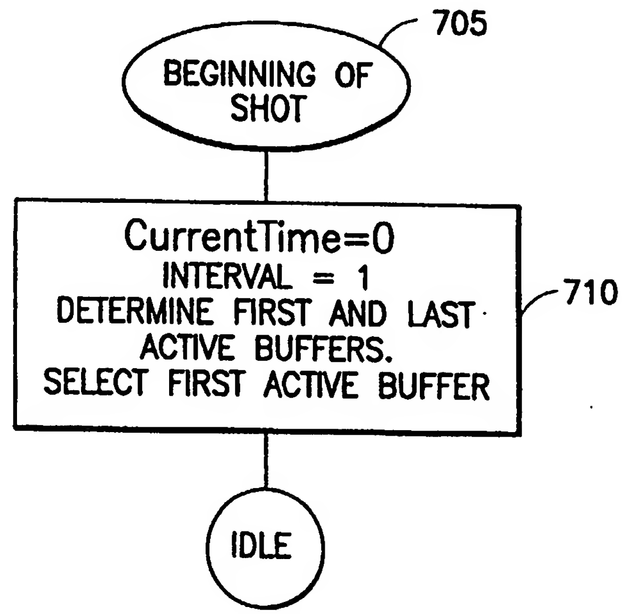

705
BEGINNING OF SHOT
CurrentTime=0
INTERVAL = 1
DETERMINE FIRST AND LAST ACTIVE BUFFERS.
SELECT FIRST ACTIVE BUFFER
710
IDLE

FIG.7B

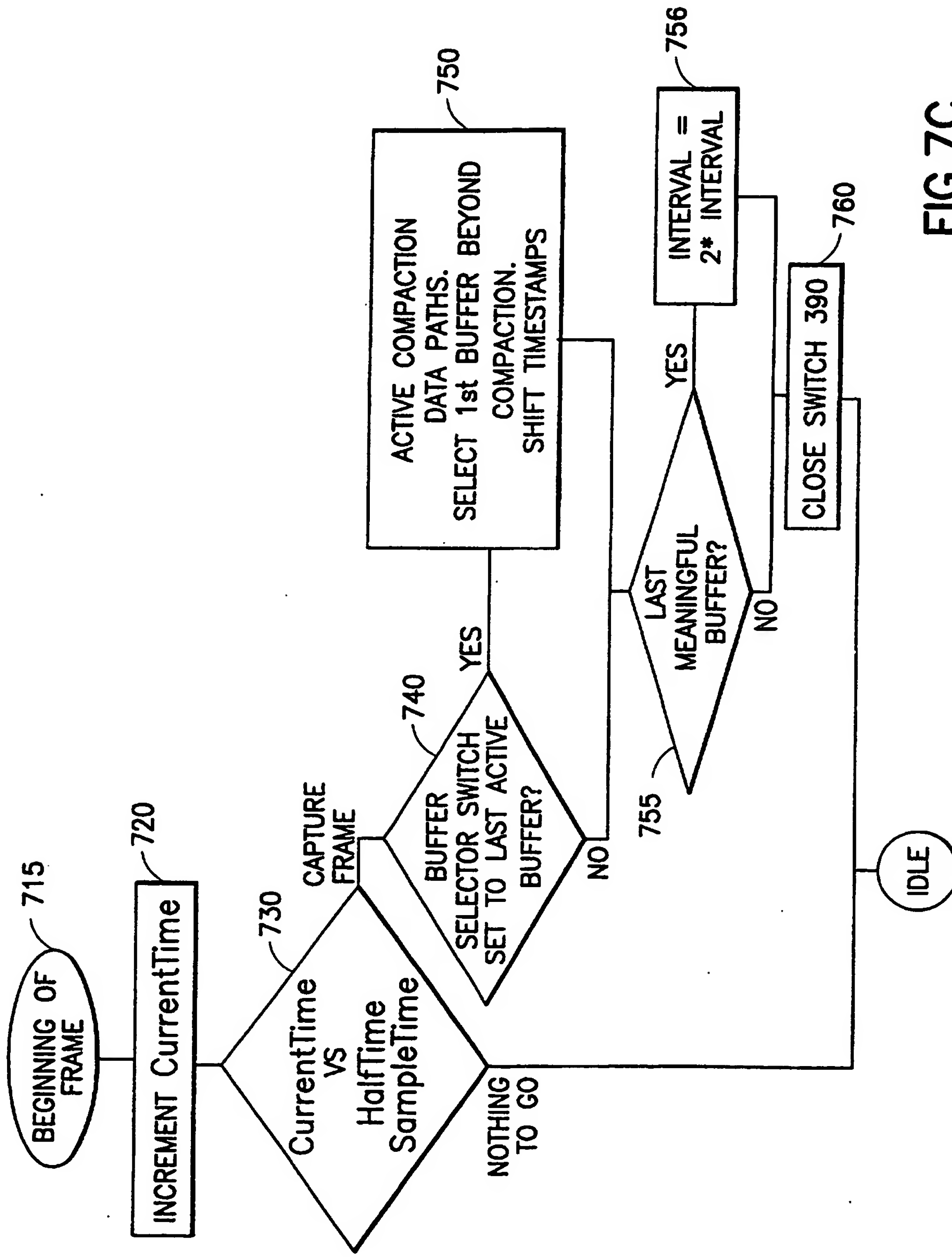
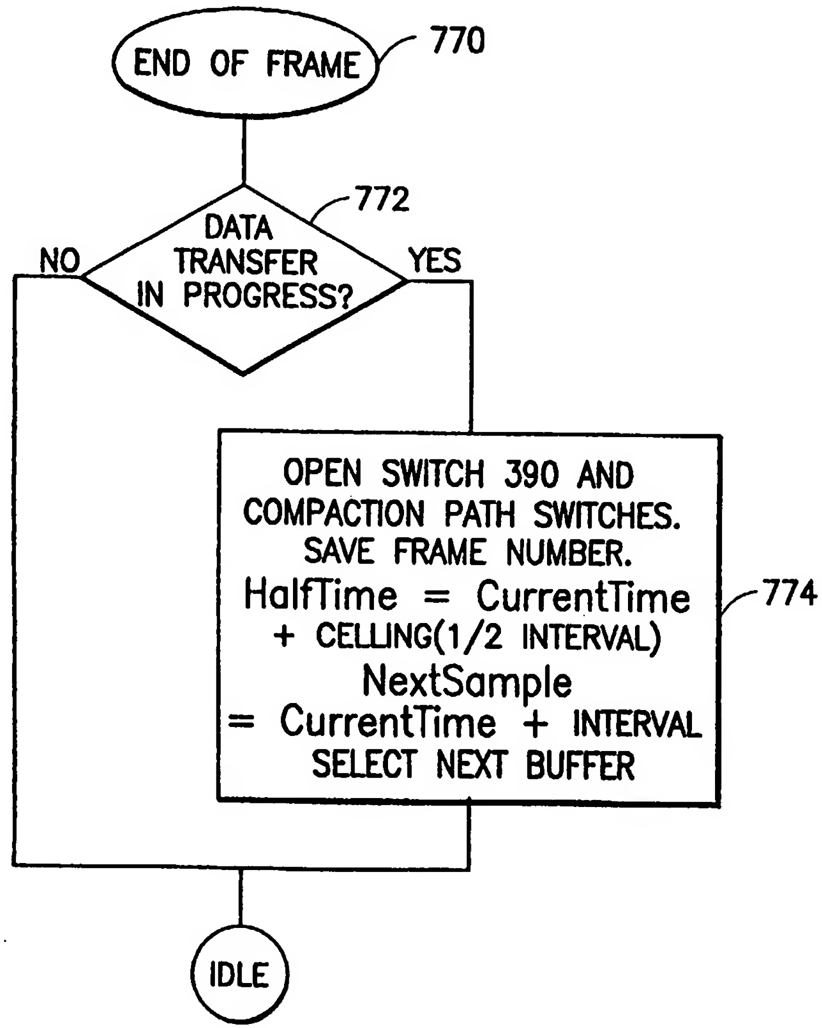
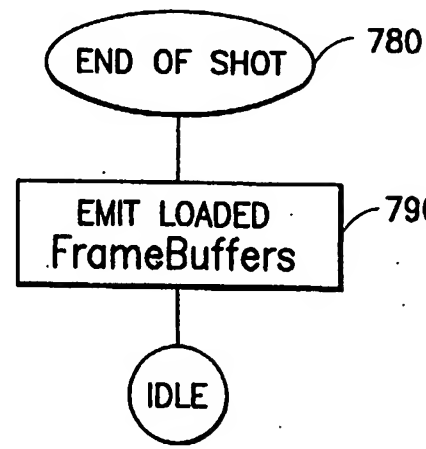

BEGINNING OF FRAME
715
INCREMENT CurrentTime
720
CurrentTime VS HalfTime SampleTime
730
CAPTURE FRAME
NOTHING TO GO
BUFFER SELECTOR SWITCH SET TO LAST ACTIVE BUFFER?
740
YES
NO
ACTIVE COMPACTION DATA PATHS. SELECT 1st BUFFER BEYOND COMPACTION. SHIFT TIMESTAMPS
750
LAST MEANINGFUL BUFFER?
755
YES
NO
INTERVAL = 2* INTERVAL
756
CLOSE SWITCH 390
760
IDLE

FIG.7C



END OF FRAME
770
DATA TRANSFER IN PROGRESS?
772
NO
YES
OPEN SWITCH 390 AND COMPACTION PATH SWITCHES. SAVE FRAME NUMBER.
HalfTime = CurrentTime + CELLING(1/2 INTERVAL)
NextSample = CurrentTime + INTERVAL
SELECT NEXT BUFFER
774
IDLE

FIG.7D



END OF SHOT
780
EMIT LOADED FrameBuffers
790
IDLE

FIG.7E

2359439

COMPUTER SYSTEM AND METHOD FOR PROCESSING AN IMAGE STREAM

This invention relates to image sequence representation and characteristic frame determination. Specifically, the invention relates to the automatic determination of a representation of an image sequence ("scene" or "shot") in terms of a sequence of selected frames or a single selected frame.

An image **stream** is one or more image **frames** played in a **sequence** to create a dynamic visual image, such as a film, videotape, video live images (from a video/film camera), visual multimedia, Magnetic Resonance Imaging (MRI), etc. (A **frame** in an image stream is one still image of that image stream, analogous to one photograph in a piece of movie film.) In general, an image stream can be any image sequence in any visual format originating from some image source, such as a video camera. An image stream can be considered to consist of multiple sequences of consecutive frames ("scenes" or "shots," hereafter referred to as **shots**. These concepts and terminology have their roots in the film industry.) The individual frames in these shots closely resemble one another because the camera location and orientation does not change, or changes only slightly, between frames of the shot. Moreover, the objects in the world that is being imaged move slowly enough, compared to the frame rate (in most cases), that the objects in the frames of the image stream move by only small amounts from frame to frame.

To produce a TV ad, movie, sitcom, or other video "product," a movie/video editor or producer goes through a process called video production or editing. In this process, the producer has to select a set of shots from a (possibly very) large library, order them, and possibly combine multiple shots into one. These video production (editing) activities include:

- browsing through previously recorded shots;

- selecting and eliminating shots;
- reordering and trimming shots; and,
- subdividing and/or uniting the shots.

To efficiently perform these activities, it is useful to represent each shot by one or more frames, where these representative frame(s) are characteristic of the whole shot. Typically, in the prior art, the representative frame(s) of the shot is either the first or last frame of the shot or both. These frames can be selected manually or automatically. One way of automatically delineating a shot (identifying the first and last frame) is by comparing adjacent frames, for instance, a pixel by pixel comparison. If two adjacent frames are very different, they are considered the last and first frames of adjacent shots.

Automatic partitioning of image streams into shots and automatic selection of characteristic frames is a desirable labor-saving convenience. Performing such a function in one pass through the image stream, or even at the time of production of the image stream, would be one time-efficient way to select representative frames.

However, the prior art fails to do this automatic selection of characteristic frames in one pass.

Prior art systems often select as characteristic frames the first and/or last frame(s) of a shot. For many shots, the beginning or ending frame may not show much and may not contain the essential event of the shot. For instance, in a shot of a person walking through the camera's field of view, the extremal frames (first and last) might not even show the person. Some frame, or series of frames, in between the first and last frame will (possibly) offer a much more illustrative sketch of the shot.

Alternatively, to obtain frames more representative of a shot, the prior art selects multiple frames of a shot, where these frames are uniformly spaced in time. There are two ways of doing this. One is to select frames at some predetermined interval (every Nth frame, for some value of N). Doing this has the drawback that if the length of the shot is not known when selecting is started, it can result in an uncontrolled number of frames being selected. The other is to calculate the selection interval, based on the desired number of resultant selected frames. However, this has the drawback that the shot length must be known at the start of selecting, in order to make the calculation. It is not possible to collect a pre-chosen number of uniformly-spaced frames from a shot whose length is not known in advance, using these methods.

Another method of prior art is content-based or event-based characteristic frame selection, where frames are selected to be characteristic if they vary by more than a threshold amount, according to some distance metric, from either the preceding frame or the preceding characteristic frame. (This technique is similar to ones often used to identify the duration of a shot, as described above.) While it can be argued that this technique produces a representative selection of frames, the temporal distribution of frames chosen in this way can be very uneven. (More frames might be chosen from one segment in the shot than from a second segment because there is more movement in the first segment.) And, as before, the final number of frames chosen is uncontrollable.

Content/event-based selecting and time-based selecting are each valid, the desirability of one versus the other depending on the specific application.

An object of this invention is an improved image stream editing system and method that obtains one or more representative frames of a sequence or shot

by processing that sequence or shot only once, without knowing its length in advance.

This object is met by the invention claimed in claim 1.

An embodiment of the present invention has one or more central processing units, one or more memories, and a buffer array with two or more buffers. Each of the buffers is capable of storing one of the shot frames. An extraction process, executed by one or more of the CPUs, selects, in the sequence, shot frames at an interframe selection interval from the image stream and stores each selected shot frame (called buffer frames) in one of the buffers so that a buffer order is maintained. The buffer order has the same order of precedence as the stream order. A culling process, executed by one or more of the CPUs, retains one or more selected shot frames (called retained selected shot frames) at a buffer interval in the buffer order and discards the remaining buffer frames. The culling also increases the interframe selection interval by multiplying the interframe selection interval by the buffer interval. A compacting process, executed by one or more of the CPUs, compresses the retained selected shot frames in the buffer order to create space in the buffer array for more selected shot frames. In this manner, the embodiment extracts a representative selection of frames by selecting some that are approximately temporally uniformly spaced throughout the shot.

An embodiment of the invention will now be described, by way of example, with reference to accompanying drawings, in which:

Figure 1 is a block diagram of an image stream editing system (workstation).

Figure 2, comprising Figures 2A and 2B, is a block diagram of uniformly spaced selected frames of a shot.

Figure 3 is a block diagram of part of the workstation's control electronics, specifically, the preferred frame extractor for selecting multiple, approximately uniformly spaced representative frames of the shot.

Figure 4A is a block diagram showing how the selected frames of the shot (buffer frames) are selected.

Figure 4B is a diagram of the relationship between the contents of four registers of the frame extractor of Figure 3 that keep track of temporal events during the extractor's operation.

Figure 5 is a block diagram of two preferred schemes, Figures 5A and 5B, for culling and compacting selected shot frames.

Figures 6a and 6b are diagrams of additional data paths in the hardware of Figure 3, configured in two alternative ways.

Figures 6c, 6d and 6e are diagrams of how reordering entries in the lookup table can achieve culling and compaction.

Figure 7 is a set of five flowcharts (Figures 7A - 7E respectively) showing various processes performed by the embodiment.

Figure 1 is a block diagram showing a video production system 100 including a video (or multimedia) editing workstation 130. One or more video sources, such as a video camera 110 or tape player 120, supply input to the workstation 130, which comprises an operator display screen 140 (which may or may not have a pointing device 142 associated with it), an operator

control panel / keyboard 150, one or more local video storage devices (e.g., tape recorder/players) 160, and workstation control electronics 170, part of which is the computer and other electronics which make up the frame extractor 175 of the present embodiment. The workstation operator (the video editor/producer) may request the workstation to process one or more video shots, extracting some number of characteristic images from each, and (for example), display these images 145 on the display screen 140. The images 145 may then be used for further processing, such as: being saved for display as stills, or used as iconic indices into a stored record of the shot, for any of the cutting, splicing, etc., operations described above.

Systems 100, without the frame extractor 175, used for processing image streams from video sources (110 and 120), are well known in the movie and TV production industry. However, by using the frame extractor 175, versions of the system 100 can be extended to other applications that can benefit from a sampled image stream that requires less storage but still gives a representative account of the events in the image stream. For example, systems using the invention could be used in the surveillance art (bank monitors, warehouse monitors, etc.) The selected frames would permit a user to view the essential parts of an image stream without having to view every frame of the stream. By doing this, the user can select only the subpart of the image stream of interest. This subpart would require less time to view, store, and/or transmit. For example, a user might require only a sub-sequence of an image stream electronically available from a video library or database, i.e., the user can select a sub-sequence of a history lesson and/or a video clip, like a movie. The frame extractor 175 can also have applications in selecting frames for use with time lapsed images/photography where the sub-part will have fewer frames than frames in the stream. Here the stream could be captured with a large number of frames and/or over a long period of time.

Figure 2, comprising Figures 2A and 2B, is a block diagram of a video sequence ("shot") 200 (Figure 2A), comprising a plurality of frames 210-225, typically 210, and a set of selected shot frames 260 (Figure 2B), comprising a plurality of (in this example, 4) frames 212, 217, 222, 225, typically 212, that are uniformly spaced in shot 200 (except 225, as will be explained) and selected by the the present frame extractor 175, and retain the stream order they had in shot 200.

The purpose of this figure is to explain the result the frame extractor 175 is intended to achieve.

The operator of workstation 130 (Figure 1) determines, before the shot 200 is presented to it, the maximum number of shot frames 210 that are to be included in the set of selected shot frames 260 upon completion of processing of shot 200. Typically this number is much less than the number of frames 210 in the shot 200.

The operation of frame extractor 175 is such that at all times, it maintains a set of selected shot frames 260 of shot 200 which is a good representation of the entire shot 200, in a sense to be explained.

The theoretically most representative set 260 would minimize the worst-case distance, i.e., the maximum distance (measured in frames, according to the stream order of shot 200) from any shot frame 210 to the nearest one of the selected shot frames 212. (In Figure 2, the worst-case distance is 2.) This suggests that, ideally:

- there should be as many selected shot frames 212 as possible, limited only by the maximum number specified by the operator. In this example, the number specified by the operator was 4, and 4 selections were made.

- the selected shot frames 212 should **span** the duration of the shot 200 as much as possible, that is, between shot frames 212 and 225 in shot 200, there should as many frames of 200 as possible, and
- the selected shot frames 212 should be as **uniformly spaced** as possible. That is, (with respect to the example of Figure 2A / 2B) just as the number of shot frames intervening in shot 200 between the first (212) and second (217) selected shot frames of set 260 is 4, the number of shot frames intervening in shot 200 between the second (217) and third (222) selected shot frames of set 260 is also 4. However, the last frame (225) forces a compromise: selecting it breaks the uniformity of spacing of the selected shots 260, since there are only 2 shot frames between 222 and 225, but not to select it would reduce both the number and the span of the selected shots 260.

Figure 3 is a block diagram showing details of the hardware of frame extractor 175. (This figure is meant to be suggestive of the overall principle of operation, and does not reflect very low-level details of the framebuffer data paths and switches, such as control logic for dual ported memory.)

The active video input 310 (selected by external means) feeds into a Video Front End 320, which generates the control signals Beginning of Frame (BOF) 321 and End of Frame (EOF) 322, which are sent to the general-purpose central processor (CPU) 330.

The front end 320, would, in the case of analog video input (such as NTSC), be a conventional sync separator circuit and digitizer. In the case of digital input (perhaps compressed, e.g., MPEG), 320 would be a bitstream demultiplexor/decoder, which would decompress the video component of each frame into a complete image bitmap.

CPU 330 includes both the logic of conventional CPU(s) (instruction fetch and decode; arithmetic/logical unit, etc) and sufficient random access memory or memories (RAM) to hold its program and the registers described herein, but not the shot frames.

The video front end 320 also passes the shot frame image data (pixels), which are routed through Transfer Enable/Inhibit Switch 390, a processor-controlled switch which is used to gate (on a frame-by-frame basis) image data to one selected buffer in the framebuffer array 340, via buffer selector switch 350. While the array 340 may actually consist of many buffers (typically 341), during processing of a given shot, only some of them may be "active", as described later, and only one of the "active" ones is the "selected" one, as determined by the setting of Switch 350. For every buffer in 340, there is also an associated timestamp register 342 which holds the time at which the selected shot frame was captured (that is, the frame count number). One implementation of 342 would be to extend each framebuffer of 340 by a few extra bytes and store the timestamp there.

Also, there is an preferred embodiment of 340 in which there are special inter-buffer data paths 345, which are shown in detail in Figure 6 below, but suppressed here for clarity. They are described in detail later.

Buffer Selector Switch 350 is controlled by Buffer Selector Register 355. In the simplest embodiment, the control is direct, as shown by the solid arrow. Register 355 should have sufficiently many bits as necessary to be able to have at least as many unique states as the maximum number of buffers in the buffer array, plus one; the extra state will indicate "all buffers full (none available)". Figure 3 shows Register 355 commanding Switch 350 to select framebuffer 2 in the array 340.

In an alternate embodiment, processor 330 may include a Lookup Table 352, which intermediates between Buffer Selector Register 355 and Buffer Selector Switch 350, permitting 355's value to be arbitrarily remapped to a different buffer number. (This is shown by the two dotted arrows.) This is explained in more detail later. Use of such a Table would obviate the need for Inter-buffer Data Paths 345.

One preferred embodiment for the framebuffer array is as semiconductor (e.g., CMOS) RAM memory. Another preferred embodiment is as sectors or tracks of read-write disk storage. In this case, Switch 350 should be interpreted as representing the concept of selection, rather than literally, as a circuit component.

Timing registers 301, 302, 303 and 304 hold unsigned integers which are frame counts. They are explained in more detail in Figure 4. Their preferred embodiment is as registers in the RAM memory of the general-purpose processor.

Figure 4A shows how the embodiment handles shots which are so long that frames continue to be presented to it even after it has already selected the specified maximum number of selected shot frames, by:

- "culling" (identifying certain framebuffers of the buffer array to be retained, and discarding and releasing all the others),
- compacting the retained ones (copying them into the freed-up lower-numbered buffers, making the higher-numbered framebuffers available for storing forthcoming frames of the shot), and
- increasing the inter-frame selection interval.

This invention is intended to cover all methods of enlarging the selection interval geometrically (multiplying by a constant factor M) and all methods of culling the buffer array by geometric contraction by the same constant

factor (retaining 1 element of M), but the preferred embodiment discussed in detail here will be limited to enlarging the selection interval by doubling it, and culling the buffer by eliminating alternate entries.

This Figure shows the history of shot 200 being sampled by the device when its operator has requested a maximum of 6 selected shot frames (in other words, to use an active framebuffer array of 6 framebuffers). The initial selection interval is 1, so as the first 6 frames (210 - 215) come along, they are selected, that is, captured into the framebuffers of the framebuffer array 340 (in this Figure and also Figure 3), whose state at this time is shown by 340A. Note that in 340A, the interframe shot distance is 1.

(The initial selection interval can be greater than 1, but the larger it is, the greater is the risk that shot 200 will end with very few selected shot frames in the framebuffer array, an undesireable result.)

The arrival of shot frame 216 forces several actions, because it is selected for storage into a framebuffer 341 of framebuffer array 340, but at this time, array 340 is full:

- **culling** of the framebuffer array 340, the result of which is that only alternate selected shot frames 210 (in other words, selected shot frames 210, 212, and 214) are **retained**.
- **compaction** of the framebuffer array 340, so the retained selected shot frames 210 that now occupy the first half of the active framebuffer array, and the second half is freed up for re-use. (The details of this are discussed in more detail later.) The state of 340 after both culling and compacting is shown in 340B. The interframe distance of the retained shot frames here (relative to the frame order of shot 200) is now 2.

* **increasing** (doubling of) the selection interval (kept in register 303, Figure 3), making it now 2 (matching the interframe distance of the retained shot frames). Shot frame 216 has already been selected, but subsequently, only every other shot frame is selected.

Normal operation (receiving shot frames as they come in, capturing the selected ones; ignoring the others) now resumes. At the time of arrival of shot frame 222, the framebuffer array 340 is in the state shown in 340C. Therefore, the same sequence of culling, compacting and doubling is performed, resulting in the state of framebuffer array 340 being as shown in 340D. The selection interval has now become 4 and the interframe distance (relative to shot 200) of the retained shot frames in 340D also becomes 4. Normal operation resumes, and continues through the arrival of shot frame 229. With the conclusion of shot frame 229, shot 200 ends, and the final state of the buffer array 340 is shown by 340E. The selected shot frames 210, etc, in it at this time are a good representation of shot 200 as explained in the discussion of Figure 2, in the sense that they span the shot, are uniformly spaced, and there are the maximal allowed number of them.

This cycle of buffer array filling, culling and compacting can be repeated almost indefinitely, to permit handling very long shots. The limiting factor is the size of the registers 301, 302, 303 and 304 in Figure 3, but even if these are 32 bits, shots can be handled that are over 4 years in length, assuming a frame rate of 30 frames/second.

There is one detail that has been glossed over in the description up till now for the sake of clarity of explanation. That is that once the selection interval elapses and a shot frame is selected, selection of the next one is performed sooner than one selection interval later; it is performed after half the selection interval (rounded down) has elapsed, and

continues with every frame following the half-interval until the full interval has expired. This is how the last frame, 229, comes to be selected. If this extra selecting were not performed, shot frame 226 would be the last one selected, and there would be a "dangling" tail of 3 shot frames at the end 200. So, once half the selection interval has elapsed, every shot frame thereafter is selected stored into a framebuffer array element, but the buffer selector 350 switch (in Figure 3) is not advanced; therefore each new shot frame overwrites the preceding one, until the full selection interval again elapses. (All such selected shot frames will be called "ephemeral".) The shot frame captured when the full selection interval elapses is not ephemeral, and after it is received, the buffer selector 350 switch is advanced. Thus, shot frame 228 is initially a selected shot frame, but it is overwritten when 229 arrives and is selected. If there had been a shot frame 230, it would have overwritten 229, but 230 would have represented the completion of a full interval (4 shot frames) since selection of shot frame 226, so it would not have been overwritten, had there been a shot frame 231.

At the beginning of shot 200, the selection interval is 1, the rounded-down half-interval is therefore 0; this would denote the frame just processed; thus no action is taken for half-interval selection when the selection interval is 1.)

If the maximum number of framebuffers to use, specified by the system's operator, is N, then the number of retained selected shot frames 210, etc, that are actually available at the end of processing Shot 200 is bounded between floor(N / 2) and N. Thus, for example, if shot 200 had ended with shot frame 221, only the 3 retained selected shot frames shown in 340D: (210, 214, 218) would have been available to represent the shot. While this is less than ideal, in terms of number of selected shot frames available, it would still be very good in terms of span of the shot (only 1

"dangling" shot frame) and fine, in terms of uniformity of spacing. While culling by a factor greater than 2 is feasible, it is not desireable because it increases the uncertainty of how many retained selected shot frames there will ultimately be.

In summary, the goal stated at the beginning of this section of obtaining a predetermined number of perfectly uniformly spaced, retained shot frames of a shot of arbitrary length that completely span it is in general unachievable, but the embodiment achieves a good approximation to it.

Figure 4B is a diagram of the relationships between the values held in four registers of the device which keep track of time. All of these values are interpreted as shot frame counts. (For the sake of explanation, they are shown with reference to a time line, but there is no explicit realization of the timeline itself in the device.) The four registers, and their values, are as follows.

- CurrentTime register 410 (301 in Figure 3) is a frame counter which is advanced by 1 each time the video front end BeginningOfFrame signal (321 in Figure 3) is received. It is initialized to 0 at start of processing shot 200.
- Interval 430 (303 in Figure 3) is the current shot inter-frame selection interval.
- HalfTime 420 (302 in Figure 3) is the time (shot frame count) at which "ephemeral" frame capture operations commence. When the value in CurrentTime is equal to or greater than the value held in this register, but less than the value of NextSample, the incoming shot frame is captured into the buffer selected by switch 350 (Figure 3), but after the capture, 350 is not advanced.
- NextSample 440 (304 in Figure 3) is the next time at which a "permanent" selected shot frame is to selected from the video source and stored into a buffer. Each time CurrentTime reaches this value,

the incoming frame is sampled into the buffer selected by selector switch 350, the switch is advanced to the next buffer, HalfTime's value is set to the sum of CurrentTime and HalfInterval, and NextSample's value is set to the sum of CurrentTime and Interval.

- PreviousSample 405 is not a register; it is the previous value held by NextSample 440.

The preferred embodiment of registers 410, 420, 430 and 440 is as RAM memory words in the CPU 330. If these have 32 bits, for instance, using conventional signed integer representation, at 30 frames/second, shots over 4 years in length can be handled.

Figure 5 shows two possible patterns of culling by geometric collapsing: for each set of M selected shot frames occupying consecutive elements of the framebuffer array, one element only is retained. Figure 5a illustrates the pattern of retaining the first member of each set; Figure 5b illustrates the pattern of retaining the last member of each set, where M could be any number less than or equal to the number of active framebuffers. As explained above, the preferred implementation is that M = 2. For illustration, the Figure is drawn for M = 4. The Figure also shows the effect of compacting.

Figure 5a shows how the preferred pattern, "retaining the first selected shot frame of each set", would cull and compact. The framebuffer array 520, just prior to culling, contains selected shot frames 521, ... , 529.. The same framebuffer array after culling, 530, retains only selected shot frames 521, 525 and 529. 540 shows the same framebuffer array after compaction. It is a property of this pattern is that the first frame of the shot is always retained.

"Compaction" refers to transferring the pixels of frames from one buffer to another ("shifting them down"). One way to achieve compacting is actually moving the pixels within the memory, pixel by pixel. An alternative technique, functionally equivalent, is renumbering the buffers (manipulating a lookup table (352 in Figure 3) which is part of selector switch 350). Lookup table manipulating can be used to both cull and compact. (Note that with either technique, the "time stamp" register associated with each frame buffer must be shifted or renumbered, to remain associated with its proper frame.)

Figure 5b shows how the embodiment would operate in the alternate culling pattern of "retaining last selected shot frame of each set". As can be seen in the Figure, different shot frames are retained (the fourth, eighth, and twelfth, rather than the first, fifth and ninth). Again, compaction could be achieved by physically moving frames between buffers or by renumbering the buffers via a lookup table. This pattern has the property that, at End Of Shot, the first retained selected shot frame is not the first frame of the shot; after repeated cullings and compactions, it can be arbitrarily many frames into the shot.

Figure 6 comprises two drawings 6a and 6b, detailing the data paths 545 used during the compaction of buffer array elements (540 in Figure 3), to achieve the compaction patterns shown in Figure 5. The data paths 645a and 645b begin at buffers that are to be retained, which are spaced M apart, where M is the set size (or equivalently, the reduction factor), in these drawings, M=2 as, in this example, buffers 3, 5, 7, etc, in buffer array 640A. They end at consecutive buffers starting with buffer 2 (as, 2, 3, 4, etc, respectively, in 640A, of Figure 6a). (Data path 646a will be explained below.)

Figures 6a and 6b show that the same set of data paths can be used to obtain either the "retain first member of set" or "retain last member of set" pattern of culling, by controlling how frames are loaded into the buffer array. If the framebuffers are loaded in order, beginning with the first framebuffer and ending with the last, as in Figure 6a, the "retain first member of set" pattern will result.

If the framebuffers are loaded in order, beginning with the Mth (in other words, treating the first M-1 buffers as if they came at the end), as in Figure 6b, the "retain last member of set" pattern will result. The data paths 645b would then begin at buffers 2, 4, 6, etc, and end at 1, 2, 3, etc, respectively. In this case, data path 646B permits retention/compaction of the last buffer of the ones which were "wrapped around". This data path exists in the "retain first" configuration as well, as 646A of Figure 6a, but it is disabled.

A given embodiment of the invention might be built possessing these data paths 645a or 645b; alternatively, no such data paths need be part of the invention, if the buffer renumbering technique is used, as explained later.

In the case where shot frames are actually moved, the values of corresponding pixels in all buffers connected to these data paths would be moved at the same time, which would also be time-synchronized with receipt of pixels of the newly-arriving selected shot frame. Thus copies do not "step on" each other, and no extra temporary working memory is required.

The highest-numbered buffer into which a retained selected shot frame is moved is ceiling(N/2) in Figure 6a and floor(N/2) in 6b. Once culling/compaction is complete, the buffer selector switch (350 in Figure 3) would be set to the next higher buffer, to prepare for receiving more selected shot frames, should there be any.

Figures 6c, 6d and 6e show how equivalent culling and compacting can be achieved by reordering entries in lookup table 352 (Figure 3). These Figures show the operation during a processing run where the desired configuration uses 6 framebuffers, a set size (compaction factor) of 2, and the "retain first of set" culling pattern. In this example, the incoming shot is at least 21 frames long.

Figure 6c shows the state of the lookup table 661 and the framebuffer array 651, after the first 6 shot frames have been selected, and the Beginning of Frame signal for frame 7 has been received. (The selection interval is 1.) The lookup table at this point has the identity transform (the first entry references framebuffer 1; the second, framebuffer 2, etc), so the selected shot frames were received into buffers in normal order (frame 1 in buffer 1, etc). The selector switch control register 655 indicates the last (logical) buffer number used (6); this is remapped by the lookup table 661 to the (actual or physical) buffer (also 6, in this case). But now culling will occur, retaining selected shot frames 1, 3 and 5 (highlighted in gray), followed by compaction and selection interval doubling.

Figure 6d shows the result after compaction. The lookup table has been modified so that buffers holding the 3 retained selected shot frames (1, 3 and 5) are listed first, preserving shot order. (Note that although compacted, they remain in their original buffers. This is the point of using buffer-renumbering.) The remaining buffers (2, 4, 6) are listed after these (order doesn't matter, since they will be overwritten with new selected shot frames). Selector switch control register 656 has been set to logical buffer 4 (since the first 3 logical buffers are holding the retained frames); this is remapped by the lookup table to physical buffer 2 of buffer array 652. Thus, when Frame 7's pixels are received, they will go into buffer 2, and when Frames 9 and 11 are selected (since the

selection interval is now 2), they will go into buffers 4 and 6, respectively, of array 652.

Figure 6e shows the result after the culling, compacting and doubling forced by the arrival of Frame 13. The 3 retained shots from 652 (1, 5, and 9) remain in the same buffers in 653 (1, 5, and 4, respectively); these buffers are listed in positions 1, 2, and 3 of the lookup table 663, respectively. The remaining buffers (2, 3, and 6) are then listed in positions 4 - 6 of the table for re-use, and selector switch control register 657 points to position 4 (mapping to buffer 2), for receipt of the next selected shot frame, Frame 13. When this is received, register 657 will be incremented, becoming 5 (thereby selecting buffer 3) for receipt of the selected shot frame after that (Frame 17, as the sampling interval now being 4); finally Frame 21 will be received into the 6th logical buffer, buffer 6.

Figure 7 is a set of flowcharts comprising the multiple frame extraction algorithm, which runs on the hardware of the frame extractor (75, in Figure 3).

When "start of shot" is signalled, Procedure 705 is executed. The system initializes itself (Step 710):

- CurrentTime is set to 0.
- Interval is set to 1.
- As determined by operator control settings, the numbers of the first and last active buffers are determined. (These numbers are the values which would be loaded into framebuffer selector switch 550 to make it select a particular buffer. These could be memory addresses, ordinal numbers, etc, or anything else, so long as the processor knows the sequence of values to load 550 with, to step sequentially through all buffers.

- Framebuffer select switch 550 is set to the first active buffer.

When Beginning Of Frame is signalled for each of the shot frames arriving from the video source, Procedure 715 is executed. The program increments CurrentTime (Step 720) and compares it to the other clock registers (Step 730). If it is equal to NextSample, or equal to or greater than HalfTime, capture of the shot frame about to come in will be required. Therefore, if all buffers are already full (as determined by the setting of Switch Control 550, tested in Step 740), culling and compaction of the buffer array concurrently with the storage of this image is enabled (Step 750).

Step 750 activates culling/compaction for embodiments with data paths 545. It closes the switches for compaction paths 545. (This is has no effect, in the buffer-renumbering embodiment.) It also resets the buffer selector switch 550 (via its control 555), so that the incoming frame is routed into the first buffer beyond the middle of the buffer array 540, that is, the lowest-numbered buffer which is not the destination of an active compaction data path. The timestamps (in registers 342 in Figure 3) of the frames which are being shifted down to lower-numbered buffers are also shifted down into the timestamp registers corresponding to these destination buffers. (The timestamps are initially set by Step 420, described below.)

If Test 755 determines that the incoming shot frame is to be captured into the highest-numbered active "meaningful" buffer (buffer whose contents will be retained in the next culling), Step 756 is performed, doubling the selection interval (the value of Interval, register 230 in Figure 4).

Finally, in Step 760, regardless of whether or not compacting will be performed, capture of the incoming shot frame is enabled, by closing Switch 590.

Procedure 770 is executed as each shot frame ends, when the End Of Frame signal is received from the video front end 320 (Figure 3). Upon receipt of this signal, if the frame just ended was not selected (which can be detected from the state of Switch 590; test 772), an early exit from this procedure is taken, otherwise Step 774 is performed: further receipt of pixels from the video source is inhibited by opening Switch 590, and further compaction, if compaction was active, is inhibited by opening the compaction data path (545 in Figure 6) switches. Or, if the buffer-renumbering technique for compaction/culling is used, processor 330 (Figure 3) switches around the entries in lookup table 352, in the manner previously explained. The frame number (available in CurrentTime, register 210 in Figure 4) is stored into the memory register (542 in Figure 3) associated with the selected framebuffer, as the timestamp for the just-received selected shot frame. And lastly, the timing registers for the next frame capture are advanced:

- HalfTime is set to CurrentTime + ceiling(½ * Interval)
- NextSample is set to CurrentTime + Interval

The buffer selector switch 550 is then advanced to the next buffer. This completes Step 774 and Procedure 770.

CLAIMS

1. A computer system for processing an image stream of one or more shots, each shot having one or more shot frames in a sequence, the sequence determining a stream order of the shot frames, the system comprising:

- means for extracting, in the sequence, shot frames at an interframe selection inverval from the image stream and storing each selected shot frame in buffers in a buffer array so that a buffer order is maintained, the buffer order having the same order of precedence as the stream order, the selected shot frames being buffer frames;
- means for culling the buffer array to retain one or more selected shot frames, called retained selected shot frames, at a buffer interval in the buffer order; and
- means for increasing the interframe selection inverval by multiplying the interframe selection interval by the buffer interval.

2. A system as claimed in claim 1, further including means for compacting the retained selected shot frames to occupy consecutively-numbered buffers in the buffer array, while retaining buffer order, to create space in the buffer array for more selected shot frames.

3. A computer system, as claimed in claim 2, wherein the culling and compacting are executed when a subset of buffers in the buffer array are full.

4. A computer system, as claimed in claim 3, wherein the subset is the entire buffer array.

5. A computer system as claimed in claim 1, where the buffer interval is 2.

6. A computer system as claimed in claim 1, wherein a first shot frame in the image stream is loaded in a first buffer of the buffer array.

7. A computer system as claimed in claim 1, wherein the interframe selection interval is initially set to 1.

8. A computer system as claimed in claim 2, wherein culling, compacting and selection interval increasing are repeated each time that the buffer array fills, the retained selected shot frames remaining after the last repetition of the culling, compacting and selection interval increasing being a set of sample frames for the image stream.

9. A computer system, as claimed in claim 8, wherein the culling and compaction are repeated until the image stream ends.

10. A computer system as claimed in claim 1, wherein after a fraction of the interframe selection interval is past, the extraction means selects each consecutive shot frame in the image stream as a sample selected shot and repeatedly overwrites the sample selected shot with the next consecutive shot frame until either the interframe selection interval is reached or the end of the image stream occurs.

11. A computer system as claimed in claim 10, where the fraction is one half.

12. A method for sampling a representative sequence of shot frames from an image stream, the image stream having a sequence of shot frames with a stream order, the method comprising the steps of:

a. extracting, in the shot, shot frames at an interframe selection interval from the image stream and storing each selected shot frame in a buffer in a buffer array so that a buffer order is maintained, the buffer order having the same order of precedence as the stream order, the selected shot frames being buffer frames;

b. culling the buffer array to retain one or more selected shot frames, called retained selected shot frames, at a buffer interval in the buffer order and discarding the remaining buffer frames;

c. increasing the interframe selection interval by multiplying the interframe selection interval by the buffer interval; and

d. compacting the retained selected shot frames in the buffer order to create space in the buffer array for more selected shot frames.

13. A method as claimed in claim 12, wherein steps a through d are repeated until the shot or image stream ends.

14. A method as claimed in claim 13, wherein some of the retained selected shot frames are discarded after the image stream ends according to a criteria.

15. A method as claimed in claim 14, wherein the criteria includes any one or more of the following:

- Number frames to be precisely a certain number,
- Maximal spanning of the shot, and
- Uniformity of spacing.

The Patent Office

INVESTOR IN PEOPLE

Application No:	GB 0030585.4	**Examiner:**	Ms Ceri Witchard
Claims searched:	1-15	**Date of search:**	11 June 2001

Patents Act 1977
Search Report under Section 17

Databases searched:

UK Patent Office collections, including GB, EP, WO & US patent specifications, in:

UK Cl (Ed.S): H4F (FEP FEX FGXX FRS)

Int Cl (Ed.7): H04N (5/44 5/915 5/783 7/01)

Other: Online: WPI, EPODOC, JAPIO TDB

Documents considered to be relevant:

Category	Identity of document and relevant passage		Relevant to claims
A	US 5847703	VSOFT LTD See column 4 lines 15-21.	-
A	US 5796439	SIEMENS See column 4 lines 25-43.	-

X	Document indicating lack of novelty or inventive step	A	Document indicating technological background and/or state of the art.
Y	Document indicating lack of inventive step if combined with one or more other documents of same category.	P	Document published on or after the declared priority date but before the filing date of this invention.
		E	Patent document published on or after, but with priority date earlier than, the filing date of this application.
&	Member of the same patent family		